



Science Arts & Métiers (SAM)

is an open access repository that collects the work of Arts et Métiers Institute of Technology researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <https://sam.ensam.eu>
Handle ID: <http://hdl.handle.net/10985/20172>

To cite this version :

Huafeng WU, Qingshun MENG, Jiangfeng XIAN, Xiaojun MEI, Christophe CLARAMUNT, Junkuo CAO - An Information Entropy Based Event Boundary Detection Algorithm in Wireless Sensor Networks - Symmetry - Vol. 11, n°4, p.537 - 2019




Any correspondence concerning this service should be sent to the repository

Administrator : scienceouverte@ensam.eu



Article

An Information Entropy Based Event Boundary Detection Algorithm in Wireless Sensor Networks

Huafeng Wu ¹, Qingshun Meng ¹, Jiangfeng Xian ^{1,*}, Xiaojun Mei ¹,
Christophe Claramunt ² and Junkuo Cao ^{3,*}

¹ Merchant Marine College, Shanghai Maritime University, Shanghai 201306, China;

hfwu@shmtu.edu.cn (H.W.); mengqingshun@stu.shmtu.edu.cn (Q.M.); xjmei94@163.com (X.M.)

² Naval Academy Research Institute, Lanveoc-Poulmic, BP 600, 29240 Brest Naval, France;
christophe.claramunt@gmail.com

³ Network and Data Center, Hainan Normal University, Haikou 571158, China

* Correspondence: xianjiangfeng0310@163.com (J.X.); cjk@hainnu.edu.cn (J.C.); Tel.: +86-1881-738-7865 (J.X.)

Received: 16 March 2019; Accepted: 11 April 2019; Published: 15 April 2019



Abstract: Wireless Sensor Networks (WSNs) have been extensively applied in ecological environment monitoring. Typically, event boundary detection is an effective method to determine the scope of an event area in large-scale environment monitoring. This paper proposes a novel lightweight Entropy based Event Boundary Detection algorithm (EEBD) in WSNs. We first develop a statistic model using information entropy to figure out the probability that a sensor is a boundary sensor. The EEBD is independently executed on each wireless sensor in order to judge whether it is a boundary sensor node, by comparing the values of entropy against the threshold which depends on the boundary width. Simulation results demonstrate that the EEBD is computable and offers valuable detection accuracy of boundary nodes with both low and high network node density. This study also includes experiments that verify the EEBD which is applicable in a real ocean environmental monitoring scenario using WSNs.

Keywords: event boundary detection; information entropy; wireless sensor networks; determining rules; ecological environment monitoring

1. Introduction

Wireless Sensor Networks (WSNs) are composed of a considerable number of low-cost, low-power and small-sized wireless sensors and have gained particular interest for many environment applications [1]. These sensor nodes are often utilized to monitor and detect real-world events like oil diffusion, fires, chemical leaks by monitoring various physical parameters such as humidity, concentration, temperature, salinity and so on. In many monitoring tasks, the core goal of WSNs is to detect and track unexpected and abnormal events in real time [2,3]. Event boundary detection is of particular importance to determine the scope of event occurrence and make preventive measures when unfortunate events happen in the environment. In many cases, an event may spread over a network and sensor-based area of an irregular shape. After a sudden incident such as oil spill pollution, timely and accurate detection of the diffusion range of events requires lots of sensor nodes to work together and consumes large amounts of energy. When compared with the analysis of the entire event area, event boundary detection is more efficient as it provides a proper view of the sensors that will be affected by broadcast messages [4].

This paper proposes a new lightweight Event Boundary Detection distributed algorithm (EEBD) to identify the real event boundary sensor nodes of a monitoring area using WSNs. Two main principles of Information Entropy, that is, uncertainty and information quantity [5], are applied to figure out the

probability of a node being a boundary node. We introduce an entropy based algorithm that determines the boundary nodes by comparing the values of entropy amongst the neighbouring nodes. A series of simulations show that the EEBD algorithm provides good precision when detecting event boundaries.

2. Related Work

Several studies have been presented to solve the problem of event boundary detection. A non-ranging event area detection algorithm to predict the event region considered as circular has been introduced [6]. In Reference [7], the authors propose a graph theory based location-free event boundary detection algorithm that can adjust both network parameters and the threshold dynamically. In Reference [8], the authors propose a distributed in-network event boundary detection algorithm using the node's local information to determine spatial and temporal evolution of event boundary. The authors in Reference [9] proposed an algorithm for detecting boundary nodes based on a combination of statistics with model classification and selection. A mathematical statistical method for edge node detection is presented in References [10,11], where the aim is to decide whether a boundary candidate node is located on the real boundary of an event region through information interaction with its neighbouring nodes.

In Reference [12], the authors take into account the spatial–temporal correlation of sensors to detect boundary nodes. A decentralized fault-tolerant event region detection method is put forward in Reference [13]. It can accurately identify fault nodes and eliminate the abnormal sensed data to avert false detection. At the experimental level, they tuned every threshold to get higher boundary fitting accuracy. In Reference [14], the authors presented a Secure Event Boundary Detection (SEBD) method utilizing local level information. This approach is highly resilient and provides a good compromise between node detection and random measurement fault. The main principle behind the SEBD approach is that a highly reliable determination rule and a system threshold are set for a node to recognize itself as a boundary node. In Reference [15], the hidden markov random field model and iterative conditions are used to calculate the event coverage in the WSNs but the parameters of the markov random model need to be trained by a large amount of sample data.

In Reference [16], a distributed and location-free boundary detection (DBD) algorithm for determining the event boundary is proposed in mobile WSNs. DBD only needs the sensed data of single node three-hop neighbours. However, DBD only considers the scenario of a mobile sensor node density less than 0.6, so it is not applicable to WSNs where all sensor nodes move in real time. In Reference [17], the authors utilized the QUDG for event boundary area detection and partition the event boundary area into disjoint dominating sets based on graph theory. Through periodic rotating the disjoint dominating sets of event boundary area in WSNs can realize the lifespan maximization of boundary sensors. Two algorithms are proposed for identifying faulty node and event boundary detection in Reference [18]. The algorithm can achieve accurate boundary node detection for various scalar sensed data values. The boundary width is fixed equal to half the communication radius of sensors. However, this is not suitable for low network densities and scenarios with a small communication radius of nodes because the number of boundary nodes increases as the network density and node radius increase. Then the communication cost of the boundary node reports the message to the base station increases. Conversely, if the number of boundary nodes is too small, it will affect the boundary node to fit the real boundary. To sum up, the existing boundary node detection algorithm is highly dependent on node sensing data and the binary decision result of a neighbour node with the calculation error is used as input data, which results in cumulative error. In addition, due to the high computational complexity and high energy consumption, the methods proposed above are not suitable for event boundary detection in mobile WSNs, such as marine WSNs.

3. Network Model and Assumptions

Let us consider a WSN with dense enough sensor nodes evenly deployed in a two-dimensional sea area of interest—called the sensor area—at the initial moment of event occurrence. We use the topology

control scheme proposed in a previous work [19]. When an event occurs, the relevant environmental parameters in the related area are likely to change, that is, the event reports a time-varying sensing signal to its neighbour region. Normally, the intensity of the signal will decrease with increasing distance from the event centre. An event area is the domain around the actual boundary of the event where the intensity of signal exceeds the given threshold value Th . Then all sensors in this sensor sea region receive radio wave signals from the event area. The proposed algorithm recommends a method for identifying boundary nodes.

We assume that sensor nodes are similar in terms of their computation, battery life and communication capability. Each node works well and has a loose synchronization. It assumes that the sensor's location information is correct since we require the right sensor's location information for the accurate detection of boundary sensor nodes. A series of prior definitions are given below:

- The term sensor area, denoted as Σ , not only refers to the geographical area covered by the WSN but also to the set of nodes in this area. We denote an event region as ω , which is the sub-region of Σ covered by an event and ϖ is the remaining region. Thus, $\Sigma = \omega + \varpi$. Hence, a sensor node $S_i \in \omega$ means it is an affected node while $S_i \in \varpi$ means it is a normal node.
- A sensor node S_i with its location information, that is $S_i(x_i, y_i)$, is considered to be a boundary node when it is on the actual boundary. Let us consider a boundary width R defined as the communication radius of the sensor S_i according to [13]. Let $N(S_i)$ denote the disk centred at node S_i with the radius R . Therefore S_i is a boundary node if $\|S_i, B\| \leq r$ where $\|S_i, B\|$ is the geographic distance between S_i and B the actual boundary. Then the event boundary $B(S_i)$ is the collection of such boundary nodes.

Suppose that the sensor data of nodes in the event area ω forms a gaussian distribution $N(\mu_1, \sigma^2)$ and the readings of other sensors in ϖ form another distribution $N(\mu_2, \sigma^2)$, where σ is small compared with $|\mu_1 - \mu_2|$. A sensor node is considered to be an affected node when its sensing information exceeds the threshold $\mu_{th} = \frac{\mu_1 + \mu_2}{2}$, while the others are called normal nodes. Let us suppose that the event boundary is a circular boundary centred on the event source. The sensor area is shown in Figure 1, the black nodes are the boundary nodes. Figure 1 on the right shows the local area of a single boundary node.

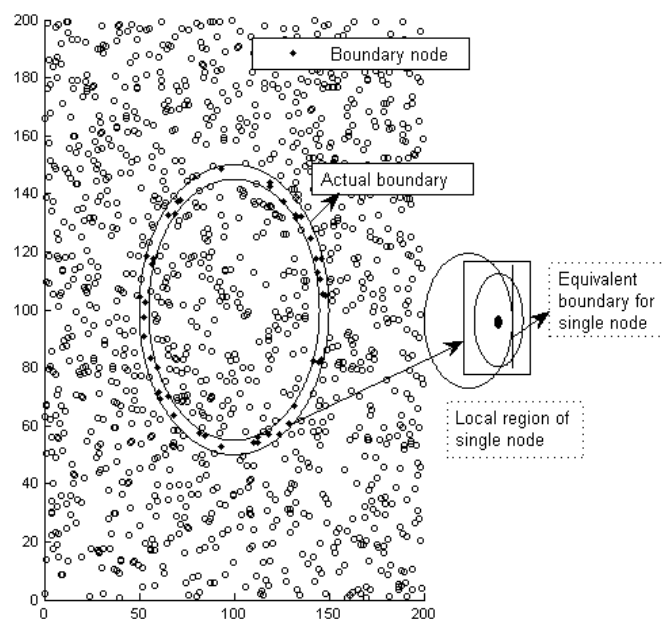


Figure 1. An illustration of a sensor area.

4. Proposed Algorithm

4.1. The Statistical Boundary Detection Model

Our goal is to design a distributed algorithm for a single node to decide whether it is a boundary sensor by comparing the sensing information with its neighbour sensor nodes. Without loss of generality let us maximize the local boundary region of a single boundary node in Figure 1, where the actual boundary can be considered a straight line. We use the nearest boundary tangent of the boundary node instead of the actual boundary of a single node. Because the event region is generally much larger than the neighbour area of a given node, the neighbour area refers to the area that a node can cover within its communication radius.

As illustrated in Figure 2, r is the boundary width and if an only if S_i lies within r , can it be considered as a boundary sensor. Let ρ denote the expected number of sensor nodes deployed per neighbour area of S_i , called network node density. Let \mathbb{N}_{S_i} denote the neighbour nodes set of sensor node S_i . The sensors in the event area are denoted as affected nodes and the remaining nodes as normal nodes.

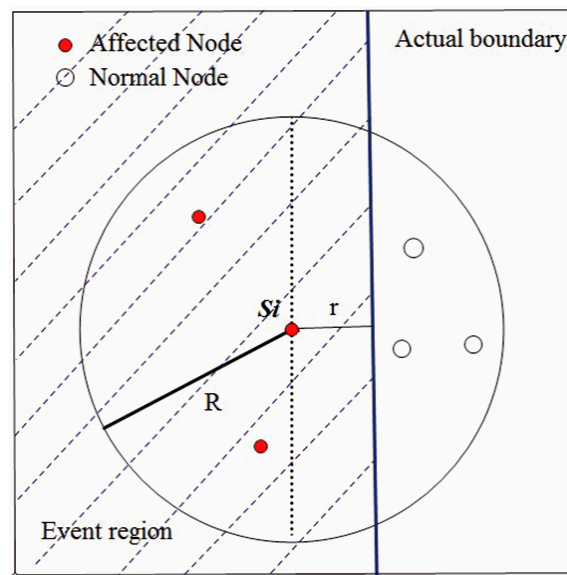


Figure 2. An illustration of Boundary Detection Model.

If a node is located exactly on the actual boundary, its neighbour nodes would have half nodes as affected nodes while the other half as normal nodes under the condition of large-scale dense sensor node deployment. In other words, such a node has a high probability of being a boundary node. However, there is a need for a determination rule to figure out whether a node belongs to $B(S_i, r)$. Thus, a proper range of proportion is needed to measure how close a node is to the actual boundary. In order to formulate this problem, let us introduce an entropy function $H(p)$ to evaluate the probability of a sensor node is a boundary sensor node. The entropy is used as a sort of probability variable that evaluates the distribution of the sensors in the neighbourhood of a given sensor. According to Reference [20], $H(p)$ is computed as

$$H(p) = -\sum_i p_i \log p_i \quad (1)$$

where p_i denotes the categories of nodes among the neighbour nodes with $i = 0, 1$. Let $i = 0$ represent the category of normal nodes, while $i = 1$ represents the affected nodes, respectively. Then p_i is computed as

$$p_i = k_i / (\rho_i - 1) \quad (2)$$

where k_i is the number of i status nodes and p_i is the network node i density. In our case, there are total two possibilities with p and $q = 1 - p$, where p denotes the ratio of affected nodes among total neighbour nodes, while q denotes the normal ones. So H can be reduced by

$$H = -(p \log p + q \log q) \quad (3)$$

which is schematized in Figure 3 as a function of p .

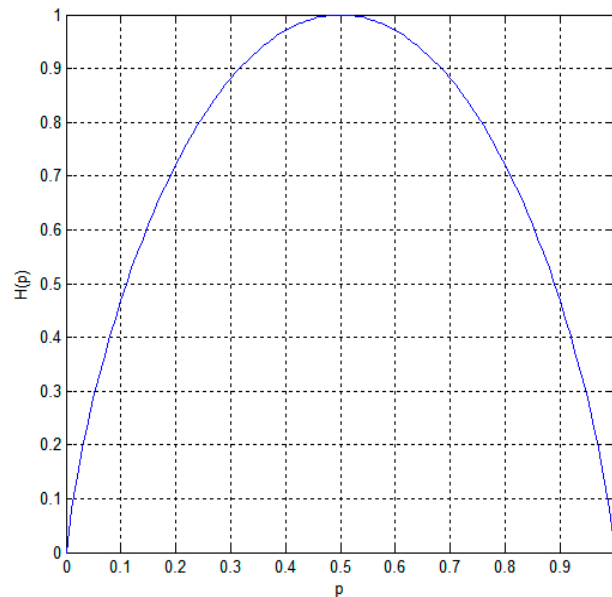


Figure 3. The characteristics of the H function.

It appears that the derived values of the entropy as denoted in (3) play an important role in calculating the probability that a sensor node can be defined as a boundary node:

1. $H = 0$ if and only if all the p_i but one are null. This means that all the neighbours of S_i are entirely affected or absolutely normal. Otherwise, if H is positive, that is, node S_i has the probability to be a boundary node. The higher the value of H is, the closer the node is to the actual boundary.
2. When all the p_i are equal (i.e., $p = q$), the value of H is the largest and is $\log 2$. This is also intuitively the higher probability a node to be a boundary node.
3. Any change towards equalization of the probabilities p, q increases H .

Figure 4 is a relationship diagram of the value of H and the number of neighbour nodes. Figure 4 shows that entropy H and the number of neighbour nodes are positively correlated.

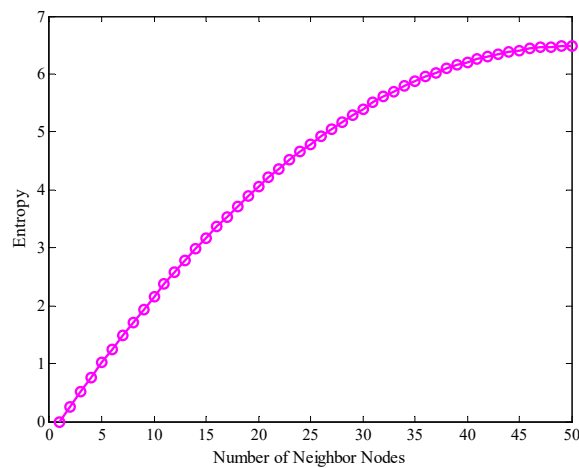


Figure 4. Entropy vs. Number of Neighbour Nodes.

4.2. Determining Rules

Consider Figure 5, where the actual boundary meets the disk of S_i at dots P_1 and P_2 while the event boundary width line meets the disk at dots P_3 and P_4 . Therefore, the actual boundary intersects the neighbouring area of S_i in two areas. Let A_1 represent the left part of \mathbb{N}_{S_i} separated by actual boundary. The remaining area of \mathbb{N}_{S_i} is represented by A_2 . So we estimate the probability of node distribution using geometric probability.

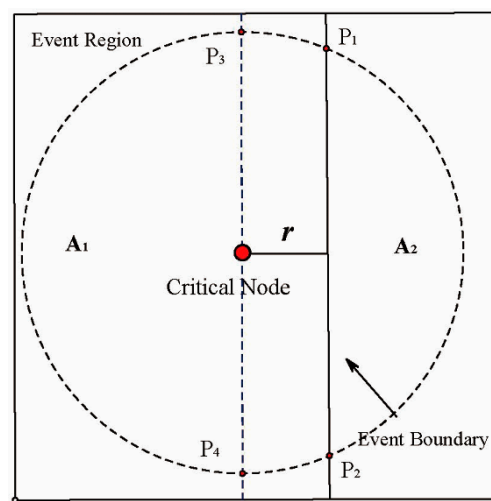


Figure 5. An illustration of critical node.

As in Figure 5, we consider a node S_i that satisfies $\|S_i, B\| = r$ as a critical node and its entropy H as the determining threshold H_{th} . A_1 has an area

$$A_1 = \frac{1}{2}\pi R^2 + S_{P_1P_2P_3P_4} \quad (4)$$

where $S_{P_1P_2P_3P_4}$ means the area of sector $P_1P_2P_3P_4$, and

$$S_{P_1P_2P_3P_4} = r\sqrt{R^2 - r^2} + R^2 \arcsin \frac{r}{R} \quad (5)$$

so that we can estimate the probability of affected nodes in \mathbb{N}_{S_i} as

$$p_{th} = \frac{1}{2} + \frac{S_{P_1P_2P_3P_4}}{\pi R^2} \quad (6)$$

Meanwhile the probability of normal nodes is

$$q_{th} = 1 - p_{th} \quad (7)$$

According to (3), the determining threshold H_{th} is

$$H_{th} = -(p_{th} \log(p_{th}) + q_{th} \log(q_{th})) \quad (8)$$

Here let us give a brief description of the case if that if the H_i of S_i beyond the H_{th} that S_i can be determined as a boundary node. Consider Figure 6, if node S_1 is in the area of $B(S_i, r)$, $p_1 < p_{th}$ and $q_1 > q_{th}$. Then $H_2 > H_{th}$. If node S_2 is outside of $B(S_i, r)$, $p_2 > p_{th}$ and $q_2 < q_{th}$. Then $H_2 < H_{th}$. In other words, nodes that fall into the area of $B(S_i, r)$ are more likely to make p and q similar, which causes the entropy to be greater than H_{th} .

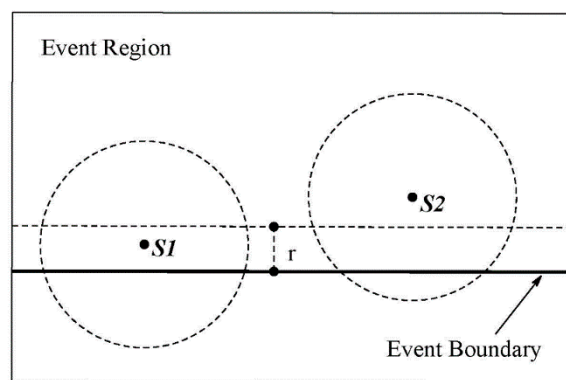


Figure 6. An illustration of boundary node and non-boundary node.

By applying the proposed detection model, we can set the boundary width r conveniently to achieve different detection goals. However, due to the geometric probability method, the relationship between r and ρ has an impact on the accuracy of the evaluation. If r is very small, H_{th} can be extremely high, that is, a few nodes can be detected as boundary nodes. If r is large, H_{th} becomes very small in some saturation H_{th} even be null as illustrated in Figure 6. Thus an appropriate r should be set to get an accurate event boundary.

Consider that nodes are deployed utilizing the grid partitioning method, then the area of a square grid δ equal to $\frac{\pi R^2}{\rho}$ where ρ is the network node density. So the grid width is $\sqrt{\delta} = R \sqrt{\frac{\pi}{\rho}}$. Firstly, the boundary region with r should contain at least one square grid δ as illustrated in Figure 7. Secondly, the boundary width r should not be greater than $R - \sqrt{\delta}$ in order not to make H_{th} too small. See examples for the above situation in Figures 7 and 8. To sum up, r should satisfy the following inequality,

$$\sqrt{\delta} \leq r \leq R - \sqrt{\delta} \quad (9)$$

so

$$R \sqrt{\frac{\pi}{\rho}} \leq r \leq R - R \sqrt{\frac{\pi}{\rho}} \quad (10)$$

As a result of the inequality, we get

$$R - R \sqrt{\frac{\pi}{\rho}} \geq R \sqrt{\frac{\pi}{\rho}} \quad (11)$$

simplification leads to $\rho \geq 4\pi \approx 13$.

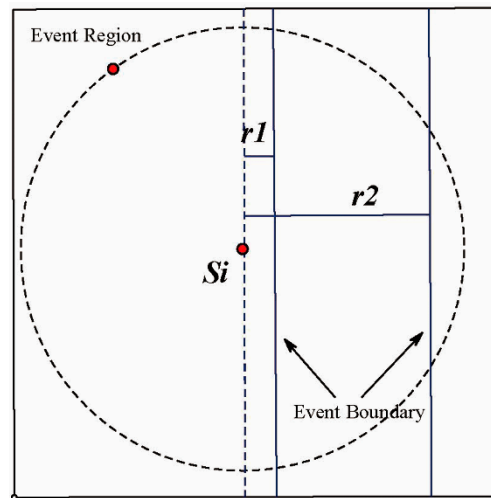


Figure 7. An illustration of extreme boundary width.

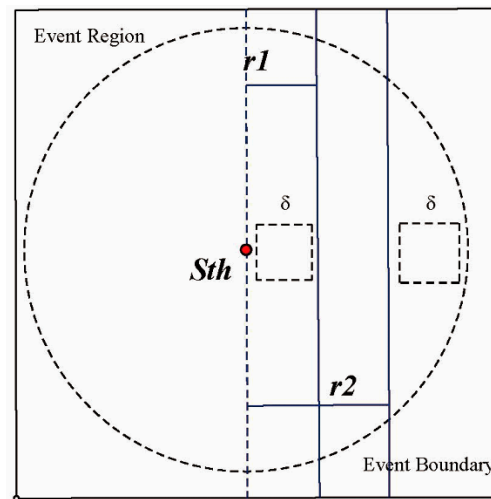


Figure 8. An illustration of the range boundary width.

From the above discussion, it can be seen that the formulation of H_{th} is influenced by the relationship between r , R and ρ . By setting the boundary width r in a certain range, we can get the appropriate threshold H_{th} and then achieve the corresponding detection results. This allows an outline that fits the event boundary to be customizable.

The EEBD algorithm consists of three steps. Firstly, each affected node S_i collects information from its neighbour nodes. Then it computes the value of H_i according to (3), that is

$$H_i = -(p \log p + q \log q) \quad (12)$$

where

$$p = \frac{\text{number of affected node}}{\text{total neighbor number}} \quad (13)$$

$$p = 1 - q \quad (14)$$

Then, sensor node S_i reports the value of H_i and listens to its neighbours node information. In the third step, given boundary width r and communication radius R to calculate the entropy threshold H_{th} which is defined in (8). If $H_i \geq H_{th}$, node S_i finally marks itself as a boundary node. The steps of EEBD algorithm are as shown below.

Algorithm 1: Entropy based Event Boundary Detection (EEBD) Algorithm

Input : Node S_i , boundary width r , node density ρ , communication radius R and $M_{S_i} = \{\}$
Output: broadcast the ID if it is the real event boundary sensor node
For each sensor node
Initialize : compute H_{th} according to (8), $H_i = 0$;
step 1;
Discover the affected sensor nodes
if sensing data $> \mu_{th}$
 $flag(i) \leftarrow 1$
else
 $flag(i) \leftarrow 0$
end
Broadcast $M_{S_i} := \{S_i, H_i, flag(i)\}$ to its neighbour nodes
if $flag(i) == 1$
 $H_i \leftarrow -(p \log p + q \log q)$
 Update M_{S_i} with new H_i
end
step 2;
Broadcast message $M_{S_i} := \{S_i, H_i, flag(i)\}$
Listen to its neighbour nodes
step 3
if $H_i \geq H_{th}$
 broadcast S_i as the final boundary node to the sink, $B(S_i, r)$

5. Simulation Results**5.1. Simulation Initialization**

In this simulation, wireless sensors are uniformly distributed in a 200×200 square area. Suppose that the square area is in the first quadrant of the plane coordinates. We suppose that the test event area is a circle with a radius of 50 and the centre of the region is located at (100, 100). Assume that normal node readings satisfy the normal distribution $N(\mu_1, \sigma^2)$ while sensor readings are depicted form $N(\mu_2, \sigma^2)$ in event area. Then the threshold $\mu_{th} = \frac{\mu_1 + \mu_2}{2}$. Mean and variance can be selected according to the actual situation, as long as the standard deviation is small compared with $|\mu_1 - \mu_2|$. In this simulation, we set $\mu_1 = 10$, $\mu_2 = 30$, $\sigma = 1$.

The communication radius is $R = 10m$. The boundary width $r = R/2$. The area δ of a grid is computed by dividing the area of a circle disk $N(S_i)$, by the network node density. The degree of fitting (DF) is applied for estimation of accuracy as a performance measure.

$$DF = \frac{|A(<r) \cap A(d)|}{|A(<r)|} \quad (15)$$

where $A(<r)$ is the node group signifying that the distance to the boundary is no more than r , while $A(d)$ is the set of boundary nodes which are detected by proposed method.

False detection rate (FR) is used to measure detection error.

$$FR = \frac{|A(>r) \cap A(d)|}{|A(>r)|} \quad (16)$$

where $A(>r)$ is the set of nodes that the distance to the boundary is beyond r . The higher DF and the lower FR mean the better performance an algorithm does.

5.2. Simulation Results

The results are averaged over 100 independently runs by NS2 (Network Simulator version 2, NS2). To evaluate the performance of EEBD, the selected benchmark algorithms are DBD [16] and LFEBD [18]. We also suppose that nodes are uniformly deployed in the monitoring region and the locations of all one hop neighbour sensors are available to a sensor node. Since this is one of the most prevalent and simple mobility models, sensor node motion is simulated using the random waypoint motion model [21]. Without loss of generality, we report the results with boundary shapes of a circle or a straight line. Figure 9a shows a visualized result of EEBD with network node density $\rho = 20$, Figure 9b presents a visualized result of EEBD with network node density $\rho = 10$. The small dots represent nodes that do not belong to the boundary sensor nodes while the bold ones represent the detected boundary sensor nodes. This shows that the detected event boundary by EEBD gives a good approximation of the event boundary as introduced in definition 3.

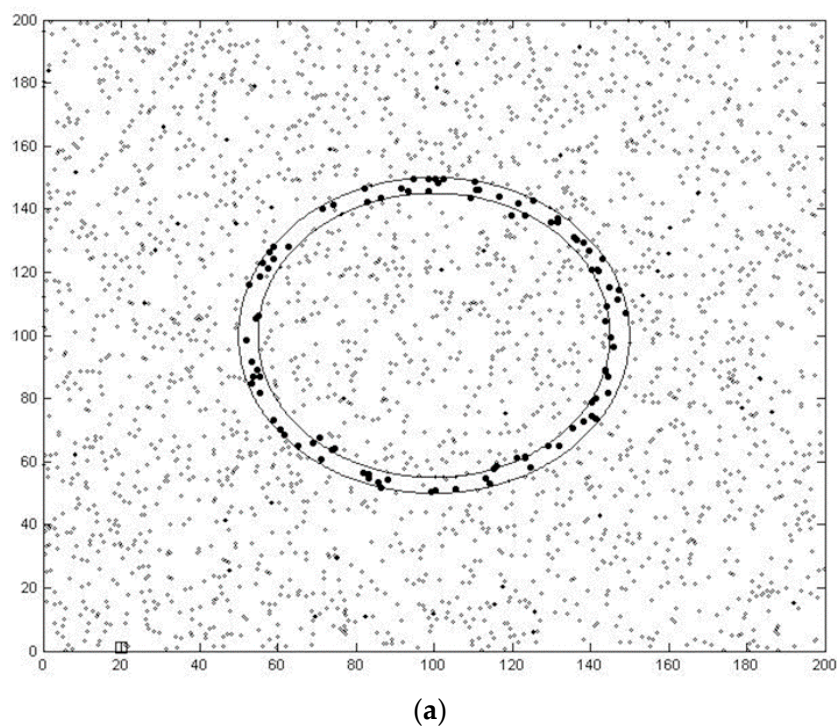


Figure 9. Cont.

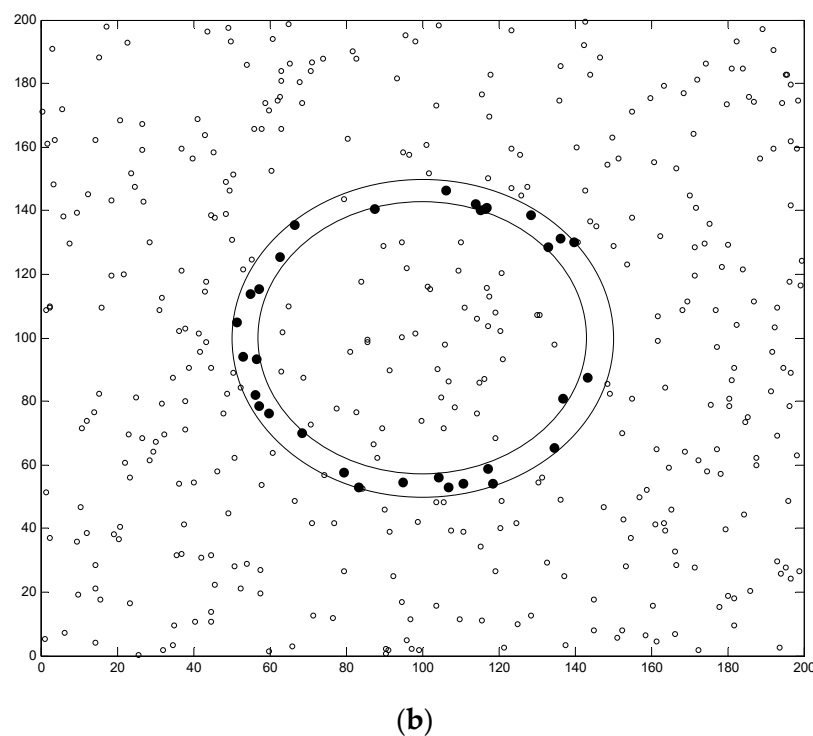


Figure 9. Visualized simulation result. (a) $\rho = 20$ (b) $\rho = 10$.

Figures 10 and 11 show comparative studies of the DF and FR, as mentioned above. From Figure 10, one can observe that the proposed algorithm has a better degree of fitting than the DBD and LFEBD algorithms when the node density is under 40. This demonstrates that the node density is proportional to the DR in both boundary shapes. When the density is increased, the number of sensors in $N(S_i)$ will increase accordingly. Thus, more sensors are close to the boundary, that is, more sensors are in the $B(S_i, r)$. So the geometric probability based on the area is more accurate. From the Figure 10, with the increase in node density, especially when it is greater than 30, both the degree of fitting increase slightly but our algorithm converges earlier with a density of about 20.

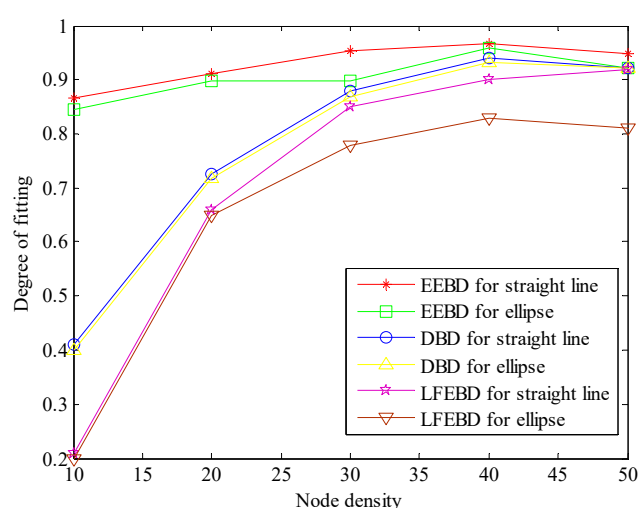


Figure 10. Degree of fitting vs. network density.

Figure 11 shows a comparative study of FR. The EEBD algorithm false detection rate gradually decreased with the increase of node density but the FR of the proposed algorithm is slightly higher than the LFEBD and DBD algorithm, especially in the case of low node density but still in a very small

range. This is owing to the small number of neighbour sensors in the case of low density, which causes the entropy of the node which is close to critical nodes to fluctuate beyond the entropy threshold but this error decreases with the increase of the node density. Thus, even if the node being misjudged is still near the boundary, this has little effect on fitting the event boundary. For practical applications, there is a certain error within the acceptable range.

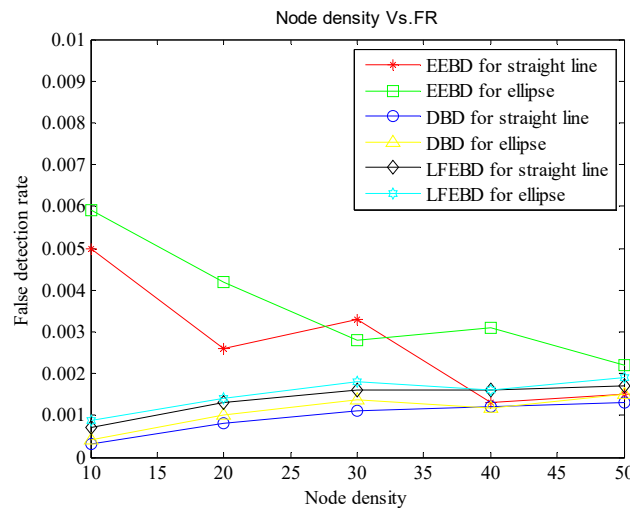


Figure 11. False detection rate vs. network density.

Figure 12 shows that setting the different boundary width has little effect on the accuracy of the detection under the constraint conditions. This permits us to try to choose few boundary nodes, both to fit the event boundary but also save data fusion and communication costs.

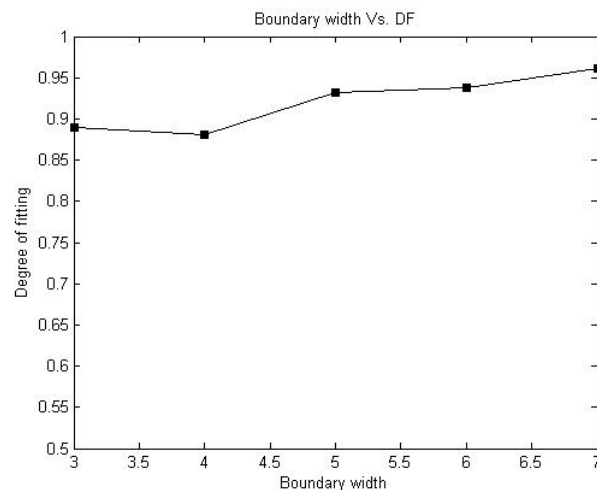


Figure 12. Degree of fitting vs. boundary width.

WSNs can be deployed on a large scale in target monitoring sea areas and are often utilized to monitor ocean information such as light, ocean currents, wind direction, hydrology and water pollution. A second series of simulations has been made in an ocean monitoring experiment using WSNs. In the marine environment, the position of the node and its relative position to the other nodes are time-varying. We randomly select 50–80% of nodes as mobile nodes and the remaining nodes are anchor nodes. The simulation set the mobile node speed from 6 to 10 m/s to demonstrate more different performances of EEBD. We use the random waypoint model to characterize the movement of the sensor nodes. A marine sensor node renews its own one-hop neighbouring node data at time intervals 5–20 s. The relationship between the detection rates and node speed are displayed in Figure 13. As the density

of the mobile node increases, the value of detection rate decreases. As can be seen from Figure 13, the updating interval greatly affects the event boundary detection rate. The relationship between updating interval and boundary detection rate is inversely proportional.

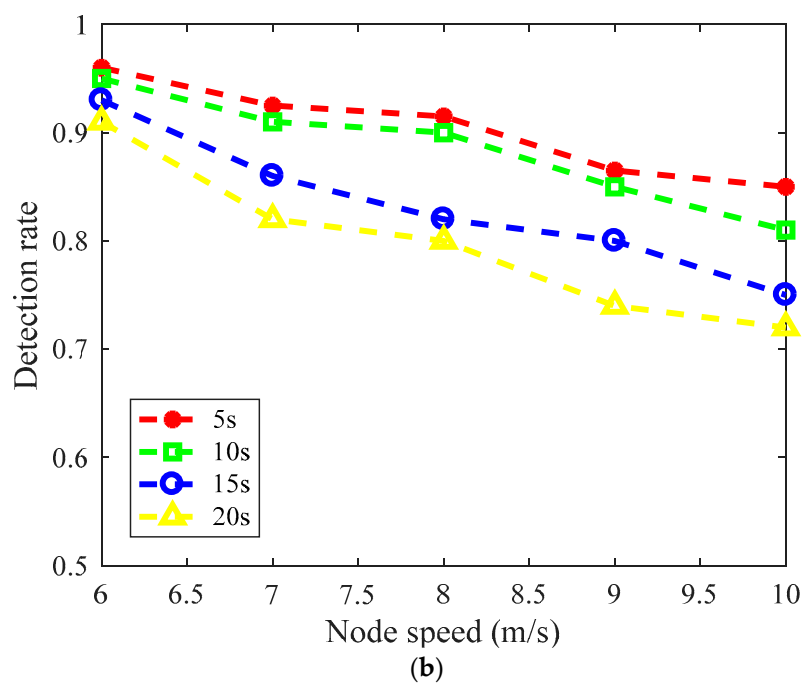
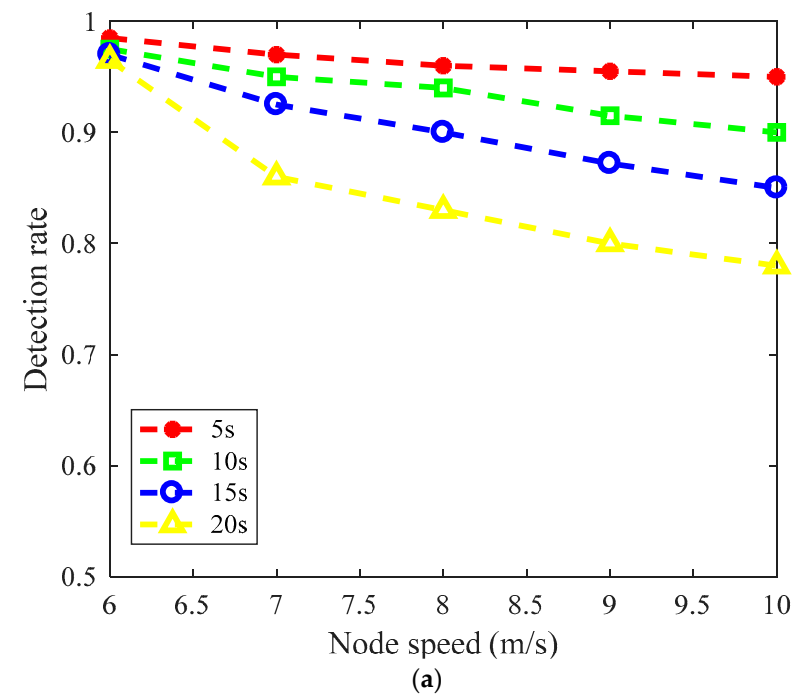


Figure 13. Cont.

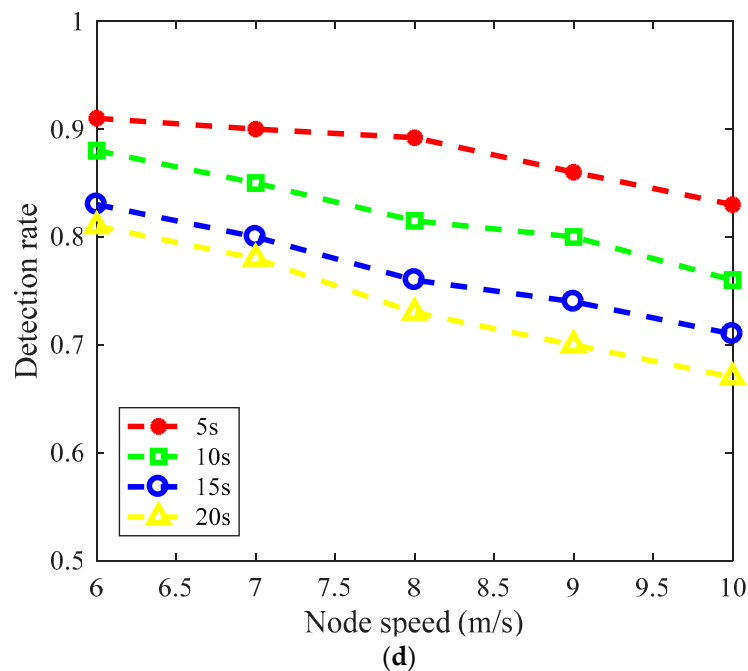
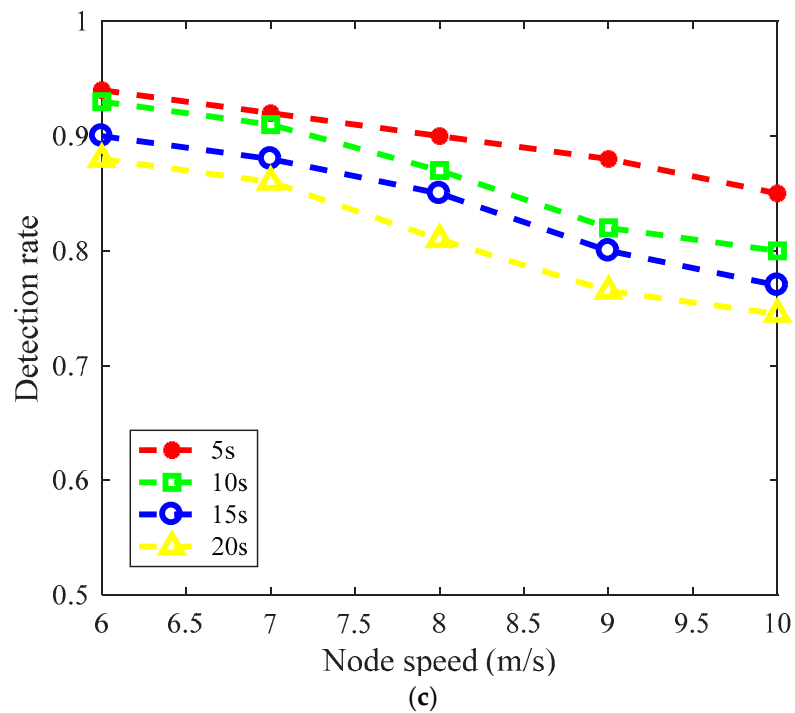


Figure 13. The boundary detection rates of EEBD with different densities of marine mobile nodes. (a) 50%. (b) 60%. (c) 70%. (d) 80%.

Figure 14 presents the computer simulation diagram of marine event boundary detection using 25 buoy nodes that integrate multiple marine sensors. Figure 15 shows the visualized simulation result of boundary detection obtained by the EEBD algorithm. As can be seen from Figure 15, the EEBD algorithm achieves accurate event boundary detection at low buoy node densities.

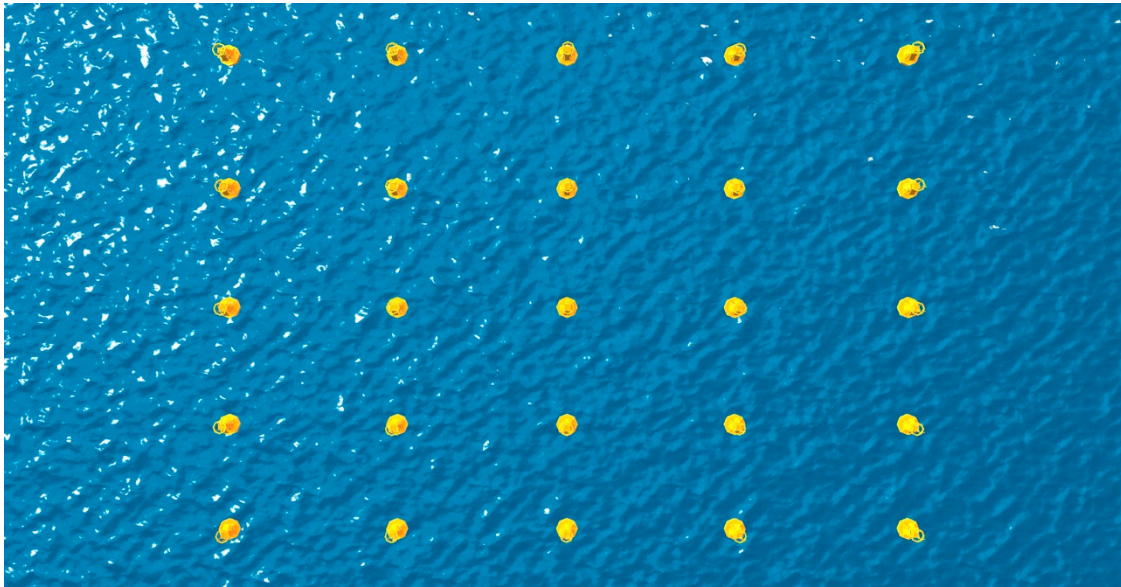


Figure 14. Marine experiment scenario diagram.

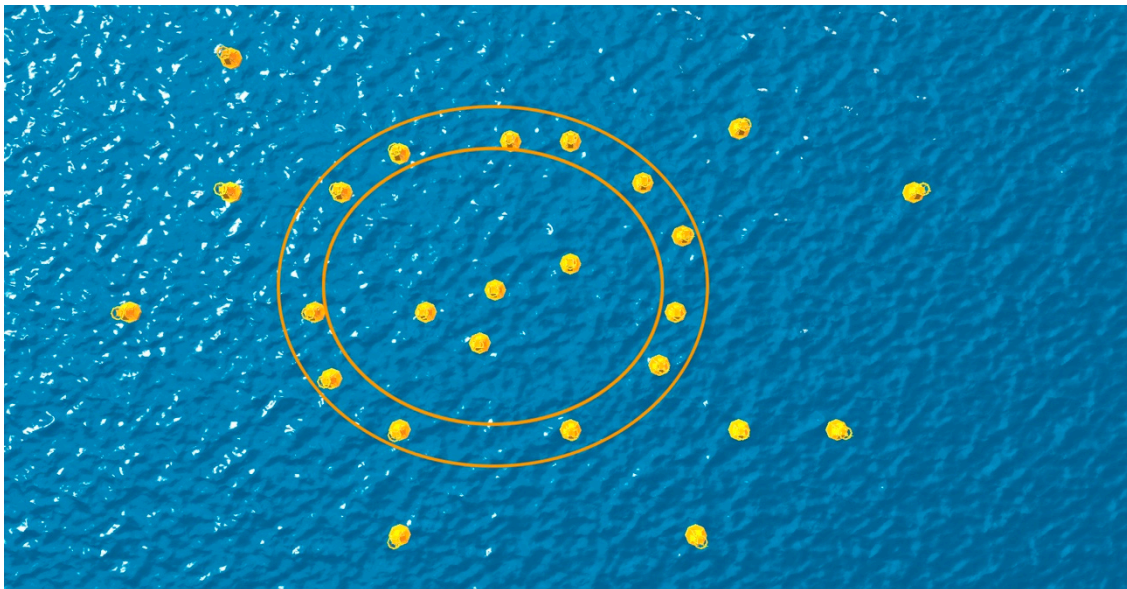


Figure 15. Visualized simulation result of marine event boundary.

5.3. Efficiency Evaluation

Overall, communication consumption is the most important energy consumption of WSNs. The communication complexity of each stage of the LFEBD is $O(d)$, because the EEBD only receives the information report of the neighbour node when the boundary detection is carried out. In the implementation of the LFEBD algorithm, nodes and neighbours need to exchange data twice, first the sensor data, secondly to evaluate the difference between the calculated data. Moreover, data exchange needs to choose a different neighbourhood. So in the detection, the proposed algorithm saves at least half of the communication consumption. In terms of computational complexity, the proposed algorithm only performs simple statistics and comparisons. In the LFEBD algorithm, each node needs to divide its neighbourhood according to the geographic coordinates into two or three parts and then update the difference, so the computational complexity of the LFEBD algorithm is higher compared to EEBD. Because each node needing its three-hop neighbouring sensing data, the computational complexity of DBD algorithm is higher compared to EEBD.

The proposed algorithm can customize the setting of the boundary width in a certain range. This allows us to control the number of boundary sensors while ensuring the boundary detection rate and reduce the communication consumption of the boundary nodes to report sensing data to the sink node.

It can be seen from the above discussion that the EEBD algorithm can achieve better detection results with less traffic and lower computational complexity, especially in low density, in the range of reasonable false alarm rate.

6. Conclusions

This experimental research introduced a distributed boundary detection algorithm EEBD. Under the same boundary detection probability, the proposed algorithm can reduce the number of nodes to be deployed compared to the two benchmark algorithms in monitoring a sea area. The boundary width can be customized to not only reduce the communication consumption of reporting but also to satisfy the detection accuracy. The simulation results demonstrate that the EEBD performs better in terms of accuracy with both low and high node density. The following two-fold directions are recommended for next study.

- (1) To further decrease the sensor nodes' energy consumption and extend network lifespan under the condition of ensuring the accuracy of event boundary detection, the energy consumption model and the event reporting routing protocol need further research.
- (2) A complete event dataset is necessary before the boundary detection algorithm is executed. Due to the adverse conditions of the sea, the collected marine big data always experience a serious data loss phenomenon in WSNs [22]. To further improve the accuracy of event boundary detection, the missing data recovery of WSNs is an important research topic.
- (3) Real marine event boundary detection experiments and event boundary dynamic tracking are also key research directions for the future.

Author Contributions: All the authors have contributed equally to this research work.

Funding: This work was supported by the National Natural Science Foundation of China [Grant No. 51579143, 51709167, 61701299]; the Shanghai Committee of Science and Technology, China [Grant No. 18040501700]; the Major Special Science and Technology Project of Hainan Province [Grant No. ZDKJ2017012]; the Graduate Innovation Foundation of Shanghai Maritime University [Grant No. 2016ycx042, 2017ycx030].

Acknowledgments: The authors would like to show their deepest gratitude for the data and technical support provided by Maritime Stereo Search and Rescue Center of Shanghai Maritime University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sheng, Z.; Mahapatra, C.; Zhu, C.; Leung, V.C. Recent Advances in Industrial Wireless Sensor Networks Toward Efficient Management in IoT. *IEEE Access* **2015**, *3*, 622–637. [\[CrossRef\]](#)
2. Minhas, U.I.; Naqvi, I.H.; Qaisar, S.; Ali, K.; Shahid, S.; Aslam, M.A. A WSN for Monitoring and Event Reporting in Underground Mine Environments. *IEEE Syst. J.* **2018**, *12*, 485–496. [\[CrossRef\]](#)
3. Wang, T.Y.; Yang, M.H.; Wu, J.Y. Distributed Detection of Dynamic Event Regions in Sensor Networks with a Gibbs Field Distribution and Gaussian Corrupted Measurements. *IEEE Trans. Commun.* **2016**, *64*, 3932–3945. [\[CrossRef\]](#)
4. Sitanayah, L.; Datta, A.; Cardell-Oliver, R. Heuristic algorithm for finding boundary cycles in location-free low density wireless sensor networks. *Comput. Netw.* **2010**, *54*, 1630–1645. [\[CrossRef\]](#)
5. Abani, N.; Braun, T.; Gerla, M. Proactive caching with mobility prediction under uncertainty in information-centric networks. In Proceedings of the 4th ACM Conference on Information-Centric Networking, Berlin, Germany, 26–28 September 2017; pp. 88–97.
6. Kundu, S.; Das, N. In-network area estimation and localization in Wireless Sensor Networks. In Proceedings of the GC'12 Workshop: The 7th IEEE International Workshop on Heterogeneous, Multi-Hop, Wireless and Mobile Networks, Anaheim, CA, USA, 3–7 December 2012; pp. 431–435.

7. Li, X.; He, S.; Chen, J.; Liang, X.; Lu, R.; Shen, S. Coordinate-free distributed algorithm for boundary detection in wireless sensor networks. In Proceedings of the Global Telecommunications Conference (GLOBECOM 2011), Houston, TX, USA, 5–9 December 2011; pp. 1–5.
8. Sadeq, M.J.; Duckham, M.; Worboys, M.F. Decentralized Detection of Topological Events in Evolving Spatial Regions. *Comput. J.* **2013**, *56*, 1417–1431. [[CrossRef](#)]
9. Ding, M.; Cheng, X. Robust event boundary detection in sensor networks—A mixture model based approach. In Proceedings of the IEEE INFOCOM 2009, Rio de Janeiro, Brazil, 19–25 April 2009; pp. 2991–2995.
10. Liao, P.; Chang, M.; Kuo, C. Distributed edge detection with composite hypothesis test in wireless sensor networks. In Proceedings of the IEEE Global Telecommunications Conference, 2004, GLOBECOM '04, Dallas, TX, USA, 29 November–3 December 2004; pp. 129–133.
11. Kundu, S.; Das, N.; Roy, S.; Saha, D. Irregular-Shaped Event Boundary Estimation in Wireless Sensor Networks. In *Progress in Intelligent Computing Techniques: Theory, Practice and Applications*; Springer: Singapore, 2018; pp. 423–435.
12. Sajida, I.; Young-Bae, K. A Continuous Object Boundary Detection and Tracking Scheme for Failure-Prone Sensor Networks. *Sensors* **2017**, *17*, 361.
13. Duh, D.; Li, S.; Cheng, W. Distributed Fault-Tolerant Event Region Detection of Wireless Sensor Networks. *Int. J. Distrib. Sens. Netw.* **2013**, *5*, 286–291. [[CrossRef](#)]
14. Ren, K.; Zeng, K.; Lou, W. Secure and fault-tolerant event boundary detection in wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2008**, *7*, 354–363. [[CrossRef](#)]
15. Dogandzic, A.; Zhang, B. Distributed Estimation and Detection for Sensor Networks Using Hidden Markov Random Field Models. *IEEE Trans. Signal Process.* **2006**, *54*, 3200–3215. [[CrossRef](#)]
16. Chu, W.C.; Ssu, K.F. Location-free boundary detection in mobile wireless sensor networks with a distributed approach. *Comput. Netw.* **2014**, *70*, 96–112. [[CrossRef](#)]
17. Shukla, S.; Misra, R.; Prasad, A. Efficient disjoint boundary detection algorithm for surveillance capable WSNs. *J. Parallel Distrib. Comput.* **2017**, *109*, 245–257. [[CrossRef](#)]
18. Ding, M.; Chen, D.; Xing, K.; Cheng, X. Localized fault-tolerant event boundary detection in sensor networks. In Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005), Miami, FL, USA, 13–17 March 2005; pp. 902–913.
19. Senouci, M.R.; Mellouk, A.; Assnoute, K. Localized Movement-Assisted Sensor Deployment Algorithm for HoleDetection and Healing. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 1267–1277. [[CrossRef](#)]
20. Ouyang, W.; Liu, Y.T.; Lin, Y.W.; Chen, Y.H. Entropy-Based Distributed Fault-Tolerant Event Boundary Detection Algorithm for Wireless Sensor Networks. In Proceedings of the IEEE International Conference on Ubiquitous Intelligence & Computing and, International Conference on Autonomic & Trusted Computing, Fukuoka, Japan, 4–7 September 2012; pp. 882–887.
21. Mitsche, D.; Resta, G.; Santi, P. The random waypoint mobility model with uniform node spatial distribution. *Wirel. Netw.* **2014**, *20*, 1053–1066. [[CrossRef](#)]
22. Wu, H.; Xian, J.; Wang, J.; Khandge, S.; Mohapatra, P. Missing data recovery using reconstruction in ocean wireless sensor networks. *Comput. Commun.* **2018**, *132*, 1–9. [[CrossRef](#)]

